Test Driving Goliath

Tom Clements

Senior Developer, On The Beach

tom-clements.com | github.com/seenmyfate

@Seenmyfate

Goliath is

an open source non-blocking asynchronous Ruby web server framework

powered by

an EventMachine reactor

a high-performance HTTP parser

Rack

Ruby 1.9 runtime (MRI, JRuby Rubinius)

High fibers

Each HTTP request within Goliath is executed in its own Ruby fiber

all asynchronous I/O operations can suspend and resume without additional code

Hello, world

```
$ gem install goliath
```

```
class HelloWorld < Goliath::API
  def response(env)
    [200, {}, "Hello, world"]
  end
end
```

Start the server

```
$ ruby hello_world.rb -sv
```

Read Later

A demo

- Single Endpoint
- Fire and forget
- Validate params are correct
- Store a url

Tree

```
$ .
    ├── Gemfile
    ├── config
    │   └── read_later.rb
    ├── read_later.rb
    └── spec
        ├── read_later_spec.rb
        └── spec_helper.rb
```

spec_helper.rb

```
Goliath.env = :test

RSpec.configure do |c|
  c.include Goliath::TestHelper, example_group: {
    file_path: /spec/
  }
end
```

spec/read_later_spec.rb

```
it 'returns OK' do
  with_api ReadLater do
    get_request(query: {url: '/test'}) do |request|
      response = Yajl::Parser.parse(request.response)
      response.should eq 'OK'
    end
  end
end
```

config/read_later.rb

```
config['db'] =
 EM::Synchrony::ConnectionPool.new(size: 20) do
  Mysql2::EM::Client.new(ENV['DB_CONFIG'])
end
```

read_later.rb

```
class ReadLater < Goliath::API

  use Goliath::Rack::Params
  use Goliath::Rack::DefaultMimeType
  use Goliath::Rack::Render, 'json'

  def response(env)
    db.aquery("INSERT INTO `articles`(`url`)
    VALUES ('#{params[:url]}')")
    [200, {}, 'OK']
  end

end
```

spec/read_later.rb

```
it "returns an error" do
  with_api ReadLater do
    get_request(query: {}) do |request|
      response = Yajl::Parser.parse(request.response)
      response['error'].should eq 'Url identifier missing'
    end
  end
end
```

read_later.rb

```
use Goliath::Rack::Validation::RequiredParam,
  {key: 'url', type: 'Url'}
```

and that's it!

- github.com/postrank-labs/goliath
- igvita.com/
- github.com/seenmyfate/read_later

gem install goliath

tom-clements.com | github.com/seenmyfate

@Seenmyfate